Boost your Hardware RE with glscopeclient

Andrew D. Zonenberg
@azonenberg

hardwear.io USA 2021
Introduction
Structure of this session

• 30 mins of intro / background
• 30 mins of interactive demo
About Me

• Ph.D CS RPI ’15
  – Did my thesis on SoC architecture for security
• IOActive since then
• Lots of GPGPU, HPC, FPGA, optimization, etc
• Started work on what is now glscopeclient around 2011
IOActive and glscopeclient

• Spare time open-source project, not IOA product
  – I’m presenting on company time, so their logo is on my slides
• Recently became stable enough for me to use at work
  – Wrote several decodes to aid embedded pentest projects
  – Hoping to make it useful to the broader community
Sneak peek before we get into details...
What is glscopeclient?

• GPU accelerated rewrite of unreleased “scopeclient”
  – New frontend with emphasis on performance and scalability
  – Based on same core: libscopehal and libscopeprotocols
• Test equipment remote control
• Waveform analysis
• Permissively licensed (3-clause BSD)
  – Interop w/ commercial tooling is an explicit goal
Release timeline

• **Prerelease:** just build current git master
• **v0.1:** First official release, 1-2 months out?
• **v0.2:** Q4 ‘21 – Q1 ‘22?
  – Lots of cleanup and portability fixes
  – More complete support of various instrument features
  – Finishing incomplete protocol decodes, more validation
  – Maybe OSX support?
• **v1.0:** who knows?
Target platforms

• Linux
  – WIP packaging for Arch, RHEL/CentOS
  – Debian packages created, working on upstreaming

• Windows
  – Already in MinGW repository
  – Alpha release of binary MSI packaging

• 64-bit x86 only (for now)
  – ARM64 planned for mid term, maybe v0.2
Unsupported platforms

• OSX
  – Need to rewrite / port most of renderer to work around graphics stack issues (y u deprecate open standard APIs?)

• Most hypervisors
  – Requires OpenGL 4.3 and compute shaders
    • No emulated GPU provides this AFAIK
  – PCIe passthrough / SR-IOV GPU should work, but untested
Architecture
Custom C++ tooling can also call the libraries directly

*Here be dragons:* no ABI stability for v0.x series!!
Dataflow

Oscilloscope → Filter graph → Display

... → Data file → Synthesis
Filter graphs

- Common DSP/multimedia architecture (like GNU Radio)
- DAG of processing blocks
Threading model

• Filter graph uses custom scheduler + OpenMP
  – Blocks with no dependencies can execute concurrently

ScopeThread → WaveformThread → UI thread

...
File Formats

• Native:
  – .scopesession format

• Import:
  – Agilent / Keysight / Rigol binary
  – CSV (with support for Digilent WaveForms metadata)
  – VCD
  – WAV

• Export:
  – Protocol dumps to CSV
Supported Hardware
• They sent me free hardware!
  – … but I haven’t had time to touch it yet 😞

• Coming soon:
  – Analog Discovery 2
  – Analog Discovery Pro 3000
  – Digital Discovery
• DSO5000
• DSO/MSO6000 (no digital channel support)
• DSO/MSO7000? (untested but probably works)
• MSOX-2000
• MSOX-3000 / 3000T
• 6000E: usable but missing a few bits
  – No advanced triggers, basic level trigger only
  – No function generator support
• 5000D: early WIP, nothing merged yet
• 3000D: most stuff
• No 2000 or 4000 series support yet, but pending
• DS1000Z
• DS1100D/E
• MSO5000
ROHDE & SCHWARZ

• RTM3000 (in progress)
- SDS2000X+ (works well, but no MSO support yet)
- SDS5000X (lightly tested)
- SDS6000X? (untested, should work)
- Early SDS1000 driver in the works, not yet merged
• All MAUI based scopes use the same command set!
  – Ultra low end (WaveAce etc) are OEM rebrands, not supported
  – *Windows CE WaveSurfers have a few quirks still

• Tested on:
  – DDA5000A
  – HDO9000
  – SDA 8Zi
  – WaveSurfer 3000*
  – WaveRunner Xi / 8000
• MSO6
• MSO5 (untested but same command set as MSO6)
• MSO4 (untested but same command set as MSO6)
Performance
Factors affecting waveform capture rate

- CPU / FPGA throughput on scope
- Interface bandwidth
  - USB2 / 100baseTX are slow
  - 1000baseT better
  - USB3 / 10GbE / PCIe best
  - Optimize for less round trips and commands
- CPU throughput on host
  - General software optimization techniques here
Scaling issues

• Most entry level scopes: $O(1)$ term dominates
  – Rigol MSO5354: can’t get >1 WFM/s at any mem depth, but respectable throughput of 48 Mbps w/ 50M points

• Higher end scopes: $O(n)$ term dominates
  – Agilent MSO6034A 1ch: 33 WFM/s @ 1K pts, 3.7 @ 1M
  – LeCroy WR8404 2ch: 40 WFM/s @ 80K pts, 3.15 @ 8M
Typical performance with shallow memory

<table>
<thead>
<tr>
<th>Model</th>
<th>CH</th>
<th>Points</th>
<th>WFM/s</th>
<th>Mbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agilent MSO6034A</td>
<td>4</td>
<td>1K</td>
<td>33.0</td>
<td>1</td>
</tr>
<tr>
<td>Keysight MSOX3104T</td>
<td>4</td>
<td>2.5K</td>
<td>2.5</td>
<td>&lt;1</td>
</tr>
<tr>
<td>PicoScope 6824E</td>
<td>8</td>
<td>100K</td>
<td>33.1</td>
<td>212</td>
</tr>
<tr>
<td>Rigol MSO5354</td>
<td>4</td>
<td>10K</td>
<td>1.0</td>
<td>&lt;1</td>
</tr>
<tr>
<td>Tektronix MSO64</td>
<td>2</td>
<td>50K</td>
<td>7.0</td>
<td>5</td>
</tr>
<tr>
<td>Teledyne LeCroy HDO9204</td>
<td>2</td>
<td>100K</td>
<td>35.0</td>
<td>112</td>
</tr>
<tr>
<td>Teledyne LeCroy WR8404M-MS</td>
<td>2</td>
<td>80K</td>
<td>40.0</td>
<td>51</td>
</tr>
</tbody>
</table>
Typical performance with longer memory

<table>
<thead>
<tr>
<th>Model</th>
<th>CH</th>
<th>Points</th>
<th>WFM/s</th>
<th>Mbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agilent MSO6034A</td>
<td>4</td>
<td>1M</td>
<td>1.0</td>
<td>32</td>
</tr>
<tr>
<td>Keysight MSOX3104T</td>
<td>4</td>
<td>2M</td>
<td>0.5</td>
<td>32</td>
</tr>
<tr>
<td>PicoScope 6824E</td>
<td>4</td>
<td>1M</td>
<td>30.5</td>
<td>1952</td>
</tr>
<tr>
<td>Rigol MSO5354</td>
<td>4</td>
<td>1M</td>
<td>0.6</td>
<td>19</td>
</tr>
<tr>
<td>Tektronix MSO64</td>
<td>4</td>
<td>500K</td>
<td>3.9</td>
<td>62</td>
</tr>
<tr>
<td>Teledyne LeCroy HDO9204</td>
<td>4</td>
<td>1M</td>
<td>5.9</td>
<td>374</td>
</tr>
<tr>
<td>Teledyne LeCroy WR8404M-MS</td>
<td>2</td>
<td>800K</td>
<td>16.5</td>
<td>211</td>
</tr>
</tbody>
</table>
Other performance considerations

• Rendering is GPU performance limited
  – More samples on screen = slower
  – 50 ms to render complete 128M point trace on RTX 2080 Ti

• Filter graph complexity
  – Sequential chains of filters can’t multithread
  – Large FIR filters or FFTs are numerically intensive
  – Availability of OpenCL / AVX2 / AVX512
Capabilities
Math / DSP

- AC couple
- Autocorrelation
- DC offset
- Deskew
- Histogram
- Moving average
- Multiply
- Subtract
- Threshold
- Up/down sample
Basic embedded

- 1-wire
- CAN
- I2C
- MIL-STD-1553
- QSPI
- SPI
- UART
Debug

- JTAG
- SWD
- SWD MEM-AP
Memory

- DDR1 command bus
- DDR3 command bus
- I2C EEPROM
- SD card cmd / data
- SPI flash
High speed serial

- CDR PLL
- 8B/10B
- 64B/66B
RF / power analysis

- Digital downconversion
- FFT
- FIR filter
  - Low / high pass
  - Band pass / notch
- Phase and frequency vs time
- Spectrogram
- Waterfall
Networking

- 10base-T
- 100base-TX
- 1000base-X
- 10Gbase-R
- Base-T autonegotiation
- GMII
- RGMII
- MDIO
Mobile

- MIPI DSI
- MIPI D-PHY
PC

- DVI
- Intel eSPI
- PCIe gen 1 / 2
  - Gen 3+ planned
- USB low / full / high
  - SS planned
Signal integrity

- CTLE
- Channel emulation
- De-embed
- Emphasis insertion/removal
- Eye pattern
- Bathtub curves
- Jitter decomposition
Signal generation

- Digital PRBS-7 / 15 / 23 / 31
- Digital to NRZ / PAM4
- AWGN
- Sine
- Step
Other features
Protocol analyzer

- Tabular display of packets
- Bidirectional sync
  - Click row to jump to packet
  - Drag timeline cursor
- Filtering
Multi scope sync

• Cascade multiple instruments on common timebase
• Simple hardware setup
  – Common reference clock
  – Trigger in / out cascade
  – Touch probes to common point to calibrate delay
• Scopes don’t have to be the same!
Getting Involved
Where to go?

- [https://github.com/azonenberg/scopehal-apps](https://github.com/azonenberg/scopehal-apps)
- IRC: #scopehal on libera.chat
- Discord: #scopehal on 1bitsquared
Acknowledgements
Industry Supporters

• Work for a scope vendor?
  – We welcome dev scopes, code contributions, and more!
• We’ve received contributions from:
Contributors

- 9names
- Alyssa ‘noopwafel’ Milburn
- Anatol Ulrich
- Andres Manelli
- Antikerneldev
- Benjamin Vernoux
- Cody Holliday
- Dave Marples
- Dominik Sliwa
- Galen Schretlen
- Francisco Sedano
- Katharina B
- Kenley Cheung
- Mike Walters
- Nash Reilly
- Pepijn De Vos
- Robin Heinemann
- Rqou
- Sam210723
- Simon Richter
- Stephanie Wilde-Hobbs
- Sylvain Munaut
- Tarunik
- Tom Verbeure
- Unai Martinez-Corral
- Willem Melching
- Whitequark
- X44203
- xzcvczcx
Questions?